

Robust Online Planning

Tamir Shazman

Robust Online Planning

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Data Science

Tamir Shazman

Submitted to the Senate
of the Technion — Israel Institute of Technology
Iyar 5786 Haifa April 2026

This research was carried out under the supervision of Associate Professor Vadim Indelman, in the Department of Data and Decision Sciences, Technion

The author of this thesis states that the research, including the collection, processing and presentation of data, addressing and comparing to previous research, etc., was done entirely in an honest way, as expected from scientific research that is conducted according to the ethical standards of the academic world. Also, reporting the research and its results in this thesis was done in an honest and complete manner, according to the same standards.

Some results in this thesis have been published as articles by the author and research collaborators in conferences and journals during the course of the author's doctoral research period, the most up-to-date versions of which being:

| |
|--|
| Tamir Shazman, Idan Lev-Yehudi, Ron Benchetrit, and Vadim Indelman. Online robust planning under model uncertainty: A sample-based approach. In <i>Proceedings of the AAAI Conference on Artificial Intelligence</i> . AAAI Press, 2026. |
|--|

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Associate Professor Vadim Indelman. Thank you for investing countless hours in our meetings and for your unwavering dedication to honing my research skills. During the most challenging phases of this journey, when I felt lost in a cloud of uncertainty, your guidance was a beacon of light that helped me navigate toward a breakthrough. Your profound devotion to your students is something I deeply admire. Galileo Galilei once noted, "You cannot teach a man anything; you can only help him find it within himself." Your mentorship did exactly that—helping me find clarity and direction when I needed it most.

Second, I owe an immeasurable debt of gratitude to my wife, Dana. I simply could not have completed this research without your endless support. Thank you for stepping up to manage our home when my schedule was overwhelming, and for always providing a patient, listening ear. The great sage Rabbi Akiva, attributing all his academic achievements to his wife's sacrifices and support, told his students: "All that is mine, and all that is yours, is hers." This rings absolutely true for us; this thesis belongs to you as much as it does to me.

Finally, I want to thank my parents, who directed me from day one to value education and study. You taught me to never accept things at face value and to always ask questions. In the words of the great Jewish physicist Albert Einstein, "The important thing is not to stop questioning. Curiosity has its own reason for existing." Thank you for nurturing that lifelong curiosity in me.

The generous financial help of the Technion is gratefully acknowledged.

Contents

List of Figures

| | |
|---|-----------|
| Abstract | 1 |
| 1 Introduction | 3 |
| 2 Related Work | 5 |
| 3 Preliminaries | 7 |
| 3.1 Robust Markov Decision Process (RMDP) | 7 |
| 3.1.1 Robust Action-Value Function Dual Form | 8 |
| 3.2 Sparse Sampling (SS) | 9 |
| 3.3 Sample Average Approximation (SAA) | 10 |
| 4 Robust Sparse Sampling (RSS) | 11 |
| 4.1 Robust Action-Value Estimation via SAA | 11 |
| 4.2 RSS Algorithm | 11 |
| 5 Theoretical Analysis of RSS | 13 |
| 5.1 Performance Guarantees | 13 |
| 5.1.1 proof sketch | 13 |
| 5.1.2 Proof | 14 |
| 5.2 Computational Complexity | 21 |
| 6 Experiments | 23 |
| 6.1 FrozenLake | 23 |
| 6.2 CartPole | 25 |
| 7 Conclusions | 29 |
| A Appendix | 31 |
| A.1 Further Implementation Details | 31 |
| A.1.1 RSS Implementation with State-Action Dependent ρ | 31 |
| A.1.2 CartPole Environment | 31 |

List of Figures

| | | |
|-----|--|----|
| 5.1 | Illustration of the RSS algorithm. At each depth d , RSS samples C successor states and recursively estimates their robust values $\hat{V}_{d-1}(s'_i)$. The robust action-value estimate $\hat{Q}_d(s, a)$ is obtained by minimizing the piecewise-linear convex function $\tilde{F}_{s,a}^{\rho,d}(\eta)$. The plot displays $F_{s,a}^\rho(\eta)$, $\hat{F}_{s,a}^\rho(\eta)$, and $\tilde{F}_{s,a}^{\rho,d}(\eta)$ in red, green, and blue, respectively, along with their corresponding minima $F_{s,a}^{\rho,*}$, $\hat{F}_{s,a}^{\rho,*}$, and $\tilde{F}_{s,a}^{\rho,d,*}$. Triangles indicate the minima, and black dots represent the breakpoints. As the number of samples C increases, both $\hat{F}_{s,a}^\rho(\eta)$ and $\tilde{F}_{s,a}^{\rho,d}(\eta)$ converge to $F_{s,a}^\rho(\eta)$. Since both $\tilde{F}_{s,a}^{\rho,d}(\eta)$ and $\hat{F}_{s,a}^\rho(\eta)$ are piecewise-linear and convex, their minima can be computed efficiently. | 14 |
| 6.1 | Visualization of the 8×8 <code>FrozenLake</code> environment, a stochastic grid-world where the agent starts in the top-left cell (S) and aims to reach the goal in the bottom-right cell (G), while avoiding hazardous holes represented by black squares. Due to the stochastic nature of the environment, the agent's actions may not always result in the intended direction. Cells adjacent to holes are marked with red squares containing question marks, highlighting regions of model uncertainty where the agent's planning model deviates from the true environment dynamics. . | 24 |
| 6.2 | Average discounted return comparison of RSS and SS (with/without access to true dynamics) under increasing noise variance in the hazard zone. Error bars denote standard error across 500 different seeds. . . . | 26 |
| A.1 | Average success rate (i.e., completing 200 steps without termination) comparison of RSS and SS (with/without access to true dynamics) under increasing noise variance in the hazard zone. Error bars denote standard error across 500 different seeds. | 33 |

Abstract

Online planning in Markov Decision Processes (MDPs) enables agents to make sequential decisions by simulating future trajectories from the current state, making it well-suited for large-scale or dynamic environments. Sample-based methods such as Sparse Sampling and Monte Carlo Tree Search (MCTS) are widely adopted for their ability to approximate optimal actions using a generative model. However, in practical settings, the generative model is often learned from limited data, introducing approximation errors that can degrade performance or lead to unsafe behaviors. To address these challenges, Robust MDPs (RMDPs) offer a principled framework for planning under model uncertainty, yet existing approaches are typically computationally intensive and not suited for real-time use. In this work, we introduce *Robust Sparse Sampling* (RSS), the first online planning algorithm for RMDPs with finite-sample theoretical performance guarantees. Unlike Sparse Sampling, which estimates the nominal value function, RSS computes a robust value function by leveraging the efficiency and theoretical properties of Sample Average Approximation (SAA), enabling tractable robust policy computation in online settings. RSS is applicable to infinite or continuous state spaces, and its sample and computational complexities are independent of the state space size. We provide theoretical performance guarantees and empirically show that RSS outperforms standard Sparse Sampling in environments with uncertain dynamics.

Chapter 1

Introduction

Markov Decision Processes (MDPs) provide a mathematical framework for modeling sequential decision-making under uncertainty, where an agent interacts with a stochastic environment to maximize cumulative expected rewards. Exact solutions to MDPs are often computationally infeasible, as they have been shown to be P-complete [22], and practical methods often resort to approximate solutions [16].

Online methods try to circumvent the complexity of computing a policy by planning online only for the current state [11, 24], making it particularly suitable for large or dynamic environments where computing optimal actions for all states in advance is infeasible. Among the most popular online planning methods are sample-based algorithms like Sparse Sampling [9] and Monte Carlo Tree Search (MCTS) [4], which approximate near-optimal decisions using limited computation at runtime.

Sparse Sampling has historical, theoretical and algorithmic significance: being the first algorithm to provide finite-time guarantees for online planning, and computational complexity scaling only by the planning horizon and approximation budget, rather than state-space size. It has inspired many practical tree-based online planning algorithms like MCTS [10, 2], popular algorithms for partially observable settings like POMCP [28] and DESPOT [30], and has recently been extended to theoretical guarantees of particle-belief approximations in POMDP planning [15, 14].

A major limitation of Sparse Sampling, MCTS and existing online planning methods is that they typically assume access to a *generative model*, i.e. a simulator that provides samples of next states and rewards. In practice, however, such models are often estimated from data and may introduce approximation errors. If these discrepancies are ignored, they can lead to poor or even unsafe decision-making [18].

Robust Markov Decision Processes (RMDPs) offer a theoretical framework to address this issue by explicitly modeling uncertainty in the transition dynamics [8, 19]. RMDPs define sets of plausible models and optimize for the worst-case within these sets, thereby guaranteeing performance robustness. However, solving RMDPs typically involves a computationally demanding min-max optimization over both policies and model perturbations, making them difficult to apply in online or large-scale settings.

To enhance scalability, various approaches have been proposed within the robust reinforcement learning (RL) community. Some methods utilize robust variants of function approximation [32], while others introduce sample-based algorithms that learn robust policies by interacting with an environment affected by model uncertainty [33, 21, 20, 5]. Although these techniques show promise, they are generally not tailored for online planning, as they aim to learn global policies across the entire state space rather than allocating computational effort to the specific decision at hand. In contrast to robust reinforcement learning, few works address the challenges of robust online planning, being limited to parametric uncertainty structures or deterministic MDPs [27, 12]. This highlights the need for robust online planning methods that are both general-purpose and theoretically grounded.

To address the challenge of online planning under model uncertainty, we adopt the RMDP framework to formalize robustness and introduce a new sample-based planning algorithm: *Robust Sparse Sampling (RSS)*. RSS extends Sparse Sampling to explicitly handle model uncertainty. To enable efficient robust decision-making in online settings, it leverages the theoretical properties and computational efficiency of the Sample Average Approximation framework [25]. We also establish theoretical performance guarantees.

This work addresses *online planning under model uncertainty* by formulating a sample-based robust planner and establishing its theoretical and empirical merits. Our main contributions are:

- **Algorithmic novelty.** We propose *Robust Sparse Sampling (RSS)*, which, to the best of our knowledge, is the first sample-based online planning algorithm that directly addresses robust MDPs while providing *finite-sample performance guarantees*. Notably, the complexity of RSS is independent of the size of the state space, making it suitable for environments with infinite or continuous state spaces.
- **Theoretical guarantees.** By leveraging the convergence theory of Sample Average Approximation technique, we derive an error bound between the value of the policy induced by RSS and the true optimal robust value function. This bound can be made arbitrarily small by appropriately setting the planning parameters.
- **Empirical validation.** Experiments on two benchmark domains demonstrate that RSS substantially reduces catastrophic failures and achieves higher empirical returns than classical Sparse Sampling when the transition dynamics are misspecified.

Chapter 2

Related Work

Robust Markov Decision Processes (RMDPs) were first introduced and explored by [8] and [19], both of which proposed dynamic programming approaches to compute the optimal robust policy. A significant limitation of these methods is that their computational complexity scales with the size of the state space, making them intractable for environments with exceptionally large or continuous state spaces. To overcome this computational bottleneck, more efficient approaches have been introduced. For example, [32] combined RMDPs with function approximation, a technique already prevalent in the RL community [23], and showed that, under certain conditions, robust value function approximation converges. More recently, sample-based robust RL approaches have been proposed to further alleviate the reliance on complete model knowledge. [33] introduced an online, sample-based framework for model-free robust reinforcement learning. They developed robust variants of Q-learning and value-function approximation, extending these standard techniques to accommodate environments with uncertain or misleading sample dynamics. Furthermore, [20] investigated robust RL using a generative model, establishing formal finite-sample complexity guarantees for learning approximately optimal robust policies across various uncertainty sets.

Robustness in *online MDP planning* has been explored in only a couple of recent studies. [27] introduced Robust Adaptive Monte Carlo Planning (RAMCP), which embeds Monte Carlo Tree Search in the Bayes-adaptive framework. That framework requires a prior over the transition model, and misspecifying this prior can harm performance; RAMCP seeks to hedge against such misspecification by computing a policy that is robust to prior errors. However, RAMCP still assumes that transition uncertainty follows a specific parametric form, which limits its applicability in settings where the dynamics are non-parametric or deviate from that model.

[12] introduced Uncertainty Adapted MCTS (UA-MCTS), an MCTS variant for deterministic MDPs that adjusts node selection based on estimated transition uncertainty. Although UA-MCTS demonstrates strong empirical performance in deterministic settings, it lacks formal robustness guarantees and does not extend naturally to stochastic environments, limiting its general applicability.

Chapter 3

Preliminaries

3.1 Robust Markov Decision Process (RMDP)

We consider a Markov Decision Process (MDP) defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, P, \gamma)$. The (possibly infinite) state space is \mathcal{S} . We assume the action space \mathcal{A} is finite. We assume a bounded reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, yet our analysis can be trivially extended to any time-dependent bounded reward. For the transition kernel P , $P_{s,a}(s')$ denotes the probability of transitioning to state s' given state s and action a . $\gamma \in [0, 1)$ is the discount factor.

During planning, the agent has access only to an approximate generative model of the transition kernel P^o , which is an estimate of the true transition model P . We assume that there exists a state-action dependent bound between the true and approximate transition kernels of the form:

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A}, \quad D(P_{s,a}, P_{s,a}^o) \leq \rho, \quad (3.1)$$

where $\rho \in [0, 1]$, and $D(\cdot, \cdot)$ is a distance metric between two probability distributions. ρ quantifies the maximum allowable deviation between the true transition model $P_{s,a}$ and the estimated model $P_{s,a}^o$. Higher values of ρ indicate greater uncertainty, with $\rho = 0$ meaning perfect model accuracy. This uncertainty bound can be estimated from statistical confidence intervals [1] or explicitly defined based on domain-specific knowledge. In this work, we focus on $D(\cdot, \cdot)$ being the Total Variation (TV) distance, i.e., $D(P_{s,a}, P_{s,a}^o) = \frac{1}{2} \|P_{s,a} - P_{s,a}^o\|_1$.

Planning directly with the empirical model P^o can lead to suboptimal or unsafe policies [18]. To guard against model error, we adopt the Robust MDP framework (RMDP). Instead of a single transition kernel, RMDP consider an uncertainty set of transition kernels. Adopting the common rectangularity assumption [8, 19], according to which the uncertainty in the transition kernels is independent for each state-action

pair, we define the uncertainty set as:

$$\mathcal{P} = \bigotimes_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mathcal{P}_{s,a}, \quad (3.2)$$

$$\mathcal{P}_{s,a} = \left\{ P_{s,a} \in \Delta(\mathcal{S}) : D(P_{s,a}, P_{s,a}^o) \leq \rho \right\}, \quad (3.3)$$

where $\Delta(\mathcal{S})$ is the set of probability distributions over \mathcal{S} . For a fixed model P' and policy π , the (non-robust) value function is

$$V^{\pi, P'}(s) = \mathbb{E}_{P', \pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_t = \pi(s_t) \right]. \quad (3.4)$$

The *robust* value function takes the worst case model in \mathcal{P} :

$$V^\pi(s) = \min_{P' \in \mathcal{P}} V^{\pi, P'}(s), \quad (3.5)$$

and our planning objective is to find a policy that maximizes this worst-case return, i.e. $V^*(s) = \max_{\pi} V^\pi(s)$ where $\pi^* \in \arg \max_{\pi} V^\pi(s)$. We denote the corresponding robust action-value function by Q^* . A deterministic robust optimal policy is known to exist [8], and its value function satisfies the robust Bellman equation:

$$V^*(s) = \max_{a \in \mathcal{A}} \left[r(s, a) + \gamma \min_{P_{s,a} \in \mathcal{P}_{s,a}} \mathbb{E}_{s' \sim P_{s,a}} [V^*(s')] \right]. \quad (3.6)$$

This formulation guarantees that the robust value serves as a lower-bound for the true value, providing explicit protection against transition-model misspecification.

3.1.1 Robust Action-Value Function Dual Form

Computing a robust policy under an imperfect transition model using online, sample-based methods is challenging, due to the infinite number of possible transition distributions within the uncertainty set (3.2). This makes a direct optimization of the robust Bellman's equation (3.6) intractable. In their recent work, [21] show that the dual form of the optimal robust action-value function admits the following closed-form expression:

$$Q^*(s, a) = r(s, a) - \gamma \min_{\eta \in \left[0, \frac{2}{\rho(1-\gamma)}\right]} \left(\mathbb{E}_{s' \sim P_{s,a}^o} [(\eta - V^*(s'))_+] - \eta + \rho \left(\eta - \inf_{s''} V^*(s'') \right) \right), \quad (3.7)$$

where $[x]_+ \triangleq \max\{0, x\}$. The dual variable η serves as a Lagrange multiplier, balancing the trade-off between the expected value and the worst-case value.

However, estimating the infimum of the robust value function $V^*(s'')$ over all states s'' is generally intractable, and particularly problematic in large or continuous state spaces, where computing the infimum term is computationally prohibitive. To simplify the dual formulation, we assume the existence of a fail-state, stated in the following

assumption.

Assumption (Fail-State). There exists a state $s_f \in \mathcal{S}$ such that $r(s_f, a) = 0$ and $P'_{s_f, a}(s_f) = 1$ for all actions $a \in \mathcal{A}$ and all transition probabilities $P' \in \mathcal{P}$. This implies $V^*(s_f) = 0$, and hence $\inf_{s''} V^*(s'') = 0$. Under this assumption, equation (3.7) simplifies to:

$$Q^*(s, a) = r(s, a) - \gamma \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} \left(\mathbb{E}_{s' \sim P_{s, a}^o} [(\eta - V^*(s'))_+] - \eta(1 - \rho) \right). \quad (3.8)$$

To simplify the notation in the remainder of the paper, we define for each state-action pair (s, a) the function:

$$F_{s, a}^{\rho}(\eta) \triangleq \mathbb{E}_{s' \sim P_{s, a}^o} [(\eta - V^*(s'))_+] - \eta(1 - \rho). \quad (3.9)$$

Hence, we can rewrite the dual action-value in equation (3.8):

$$Q^*(s, a) = r(s, a) - \gamma \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} F_{s, a}^{\rho}(\eta). \quad (3.10)$$

3.2 Sparse Sampling (SS)

Sparse Sampling (SS) [9] is a model-based online planning algorithm assuming a known transition kernel P , that approximates the optimal action-value function $Q^{*, P}$ with high probability by constructing a stochastic lookahead tree of finite depth H . It operates by building a recursive search tree. At each node corresponding to a state s , the algorithm explores each action $a \in \mathcal{A}$ by drawing C independent next-state samples from $P^{s, a}(\cdot)$. For each sampled successor state s' , the process recursively continues until the maximum depth H is reached. The recursive computation of the action-value function at depth d proceeds as follows:

$$\begin{aligned} \hat{Q}_d^P(s, a) &= r(s, a) + \gamma \cdot \frac{1}{C} \sum_{i=1}^C \hat{V}_{d-1}^P(s'_i), \quad s'_i \sim P_{s, a}, \\ \hat{V}_{d-1}^P(s) &= \max_{a \in \mathcal{A}} \hat{Q}_{d-1}^P(s, a), \\ \hat{V}_0^P(s) &= \tilde{V}_{\theta}^P(s), \quad \forall s \in \mathcal{S} \text{ (leaf terminal value)}. \end{aligned} \quad (3.11)$$

Here, $\tilde{V}_{\theta}^P(s)$ denotes a terminal value estimator for $V^*(s')$, which may be a learned function or set to zero. In this work, unless stated otherwise, we assume $\tilde{V}_{\theta}^P(s) = 0$.

Sparse Sampling provides theoretical guarantees on the gap between the nominal value of the policy it computes and the optimal value function. This difference can be made arbitrarily small by choosing a sufficiently large number of samples C and planning depth H .

3.3 Sample Average Approximation (SAA)

Stochastic programming [7] addresses optimization problems under uncertainty, where the objective function involves an expectation over a random variable. The general formulation is given by:

$$\begin{aligned} & \min_{x \in \mathbb{X}} F(x), \\ & \text{where } F(x) \triangleq \mathbb{E}_{y \sim P_y} [f(y, x)], \end{aligned} \tag{3.12}$$

Here, P_y denotes a probability distribution over random variables y , $\mathbb{X} \subseteq \mathbb{R}$ is the feasible domain, and $f(y, x)$ is a real-valued function depending on both the uncertain variable y and the decision variable x .

Computing the expectation $\mathbb{E}_{y \sim P_y} [f(y, x)]$ exactly can be challenging, especially when the distribution P_y is high-dimensional or analytically intractable. Sample Average Approximation (SAA) [25] replaces the expectation with an empirical average based on a finite number of samples drawn from P_y . Given C i.i.d. samples $\{y_i\}_{i=1}^C$ from P_y , the empirical approximation of the objective becomes:

$$\hat{F}(x) = \frac{1}{C} \sum_{i=1}^C f(y_i, x). \tag{3.13}$$

SAA is widely used across domains such as operations research, finance, and machine learning to address optimization problems under uncertainty [3, 26]. Its convergence properties are well-established [29]; under suitable regularity conditions on the function $f(y, x)$ and the feasible set \mathbb{X} , the solution of the empirical problem converges to the true optimum of the original stochastic program as the number of samples increases [25].

Chapter 4

Robust Sparse Sampling (RSS)

4.1 Robust Action-Value Estimation via SAA

Although the simplified dual formulation in Equation (3.10) offers valuable theoretical insight, it remains intractable to solve directly when the robust value function $V^*(\cdot)$ is unknown. Even if $V^*(\cdot)$ were available, evaluating $Q^*(s, a)$ would still require solving a stochastic programming problem over the function $F_{s,a}^\rho(\eta)$, which involves an intractable expectation, particularly in large or continuous state spaces.

To address this challenge, we employ the SAA method, replacing $F_{s,a}^\rho(\eta)$ with an empirical estimate $\hat{F}_{s,a}^\rho(\eta)$ based on C samples of the next states $\{s'_i\}_{i=1}^C \sim P_{s,a}^o(\cdot)$. This leads to the following approximate formulation:

$$\begin{aligned} \hat{Q}^*(s, a) &= r(s, a) - \gamma \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} \hat{F}_{s,a}^\rho(\eta), \quad \text{where} \\ \hat{F}_{s,a}^\rho(\eta) &= \frac{1}{C} \sum_{i=1}^C (\eta - V^*(s'_i))_+ - \eta(1 - \rho). \end{aligned} \tag{4.1}$$

The function $\hat{F}_{s,a}^\rho(\eta)$ is piecewise linear and convex in η , with non-differentiable break-points occurring at the sampled values $\{V^*(s'_i)\}_{i=1}^C$. This structure makes the optimization problem in Equation (4.1) efficiently solvable.

4.2 RSS Algorithm

The Robust Sparse Sampling (RSS) algorithm, inspired by the Sparse Sampling algorithm, incorporates robustness against model uncertainty while using a finite number of samples online in a recursive manner. Instead of estimating the nominal action-value function $Q^{*,P}(s, a)$, RSS estimates the robust action-value function $Q^*(s, a)$. The complete procedure is described in Algorithm 4.1.

Specifically, RSS recursively estimates the robust action-value function at depth d by sampling C successor states from the estimated generative model $P_{s,a}^o$. Then, in contrast to the standard Sparse Sampling, RSS computes the robust action-value

Algorithm 4.1 Robust Sparse Sampling (RSS)

Input: Current state s , current depth d **Parameter:** Sample width C , planning horizon H computed based on Theorem 5.1**Output:** Estimated optimal action and its value

```
1: if  $d = 0$  then
2:   return 0
3: end if
4: for all  $a \in \mathcal{A}$  do
5:    $V_{\text{list}} \leftarrow []$ 
6:   for  $i = 1$  to  $C$  do
7:     Sample  $s'_i \sim P^o(\cdot | s, a)$ 
8:      $(\_, \hat{V}_{d-1}(s'_i)) \leftarrow \text{RSS}(s'_i, d - 1)$ 
9:     Append  $\hat{V}_{d-1}(s'_i)$  to  $V_{\text{list}}$ 
10:  end for
11:  Update  $\hat{Q}_d(s, a)$  using Equation (4.2) with  $V_{\text{list}}$ 
12: end for
13: return  $\arg \max_{a \in \mathcal{A}} \hat{Q}_d(s, a), \max_{a \in \mathcal{A}} \hat{Q}_d(s, a)$ 
```

function by solving the SAA problem at each depth d :

$$\begin{aligned} \hat{Q}_d(s, a) &= r(s, a) - \gamma \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} \tilde{F}_{s,a}^{\rho,d}(\eta), & \text{where} \\ \tilde{F}_{s,a}^{\rho,d}(\eta) &= \frac{1}{C} \sum_{i=1}^C (\eta - \hat{V}_{d-1}(s'_i))_+ - \eta(1 - \rho), \\ \hat{V}_{d-1}(s) &= \max_{a \in \mathcal{A}} \hat{Q}_{d-1}(s, a), \\ \hat{V}_0(s) &= 0, \quad \forall s \in \mathcal{S} \text{ (leaf terminal value)}. \end{aligned} \tag{4.2}$$

where each successor s'_i is drawn i.i.d. from the approximate model $P_{s,a}^o(\cdot)$. The routine is invoked recursively from the current state s and remaining depth d . The recursion terminates at $d = 0$, where the leaf value is fixed at 0. .

It is important to emphasize that the function $\hat{F}_{s,a}^{\rho}(\eta)$ in Equation (4.1) is a theoretical construct, as it depends on the true robust value function $V^*(\cdot)$, which is not accessible to the algorithm in practice. In contrast, the RSS algorithm avoids this dependency by estimating robust values recursively.

At depth d , RSS replaces $V^*(s')$ with $\hat{V}_{d-1}(s')$, the estimated robust value of the sampled successor state s' from the previous depth. This substitution yields an empirical estimate $\tilde{F}_{s,a}^{\rho,d}(\eta)$ that approximates $F_{s,a}^{\rho}(\eta)$ without requiring knowledge of the exact robust value function.

Chapter 5

Theoretical Analysis of RSS

5.1 Performance Guarantees

Our main theoretical result establishes that the value of the policy returned by the RSS algorithm can be made arbitrarily close to the optimal robust value function. Specifically, RSS guarantees the following bound:

Theorem 5.1. *For any $s \in \mathcal{S}$ and any $\epsilon > 0$, the Robust Sparse Sampling algorithm returns a policy π such that:*

$$|V^\pi(s) - V^*(s)| \leq \epsilon,$$

with the following hyperparameters:

$$\begin{aligned} \lambda &= \frac{\epsilon}{3}, \\ \delta &= \lambda(1 - \gamma), \\ H &= \lceil \log_\gamma(\lambda) \rceil \\ C &= \frac{2}{\lambda^2 \rho^2 (1 - \gamma)^2} \cdot \left(2H \ln \left(\frac{2|A| \cdot H}{\lambda^2 \rho^2 (1 - \gamma)^2} \right) + \ln \left(\frac{2(8 - 4\rho)}{\delta \lambda (1 - \gamma) \rho} \right) \right). \end{aligned}$$

Similar result was originally shown for the nominal (non-robust) setting by [9], where the Sparse Sampling algorithm approximates the optimal value function $V^{*,P}(s)$. Here, we extend that result to the robust setting.

This extension is non-trivial, as the robust formulation must account for worst-case transitions within an uncertainty set, which do not arise in the nominal case.

5.1.1 proof sketch

First, we show that both $F_{s,a}^\rho(\eta)$, defined in (3.10), and its empirical counterpart $\hat{F}_{s,a}^\rho(\eta)$, defined in (4.1), are Lipschitz continuous with respect to η (Proposition 5.1.1). This property allows us to apply concentration inequalities from SAA theory [25], yielding probabilistic bounds between $F_{s,a}^\rho(\eta)$ and $\hat{F}_{s,a}^\rho(\eta)$ (Lemma 5.1.2). Consequently, we

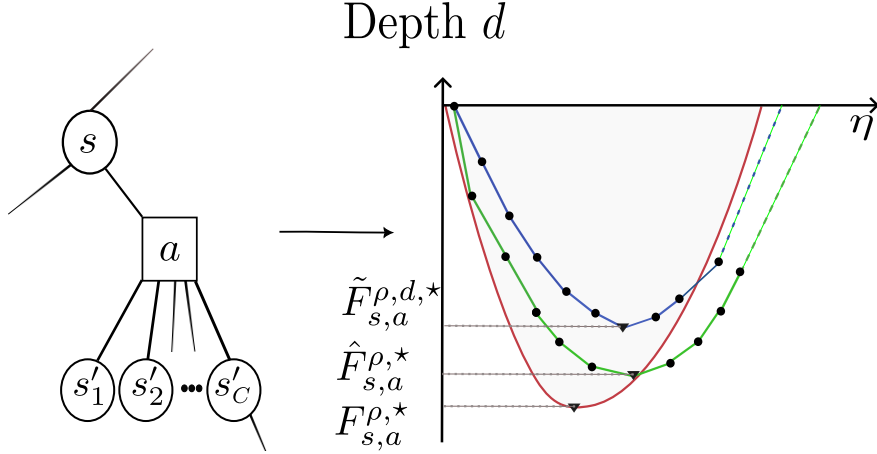


Figure 5.1: Illustration of the RSS algorithm. At each depth d , RSS samples C successor states and recursively estimates their robust values $\hat{V}_{d-1}(s'_i)$. The robust action-value estimate $\hat{Q}_d(s, a)$ is obtained by minimizing the piecewise-linear convex function $\tilde{F}_{s,a}^{\rho,d}(\eta)$. The plot displays $F_{s,a}^\rho(\eta)$, $\hat{F}_{s,a}^\rho(\eta)$, and $\tilde{F}_{s,a}^{\rho,d}(\eta)$ in red, green, and blue, respectively, along with their corresponding minima $F_{s,a}^{\rho,*}$, $\hat{F}_{s,a}^{\rho,*}$, and $\tilde{F}_{s,a}^{\rho,d,*}$. Triangles indicate the minima, and black dots represent the breakpoints. As the number of samples C increases, both $\hat{F}_{s,a}^\rho(\eta)$ and $\tilde{F}_{s,a}^{\rho,d}(\eta)$ converge to $F_{s,a}^\rho(\eta)$. Since both $\tilde{F}_{s,a}^{\rho,d}(\eta)$ and $\hat{F}_{s,a}^\rho(\eta)$ are piecewise-linear and convex, their minima can be computed efficiently.

obtain bounds on the difference between the true robust action-value function $Q^*(s, a)$ and the SAA-based estimate $\hat{Q}^*(s, a)$. Next, we establish a concentration bound between $F_{s,a}^\rho(\eta)$ and $\tilde{F}_{s,a}^{\rho,d}(\eta)$, the estimator used by RSS at depth d , as defined in (4.2) (Lemma 5.1.3). This step uses the bound from the previous stage, combined with an inductive argument over the estimated robust value function \hat{V}_{d-1} at depth $d-1$. The relationship and differences between $F_{s,a}^\rho(\eta)$, $\hat{F}_{s,a}^\rho(\eta)$, and $\tilde{F}_{s,a}^{\rho,d}(\eta)$ are illustrated in Figure 5.1.

We then apply a union bound over every state-action pair in the search tree, guaranteeing that the concentration inequalities hold simultaneously at all nodes.

Finally, to relate the robust value of the policy $V^\pi(s)$ returned by RSS to the optimal robust value $V^*(s)$, we generalize a key lemma from [9] to the robust setting. This lemma bounds the value gap in terms of the maximum approximation error in the robust action-value function. Combining all steps yields the final bound stated in Theorem 5.1.

5.1.2 Proof

We first analyze functions $F_{s,a}^\rho(\eta)$ and $\hat{F}_{s,a}^\rho(\eta)$ defined in Equations (3.10) and (4.1), respectively.

Proposition 5.1.1. *For any state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, the functions $F_{s,a}^\rho(\eta)$ and $\hat{F}_{s,a}^\rho(\eta)$ are convex and $(2 - \rho)$ -Lipschitz continuous with respect to η .*

Proof. We first show that the function is Lipschitz w.r.t η . For $\eta_1, \eta_2 \in \left[0, \frac{2}{\rho(1-\gamma)}\right]$, we have:

$$\begin{aligned} |F_{s,a}^\rho(\eta_1) - F_{s,a}^\rho(\eta_2)| &\leq (1-\rho) \cdot |\eta_1 - \eta_2| + \\ &\left| \mathbb{E}_{s' \sim P_{s,a}^\rho} [(\eta_1 - V^*(s'))_+ - (\eta_2 - V^*(s'))_+] \right| \end{aligned} \quad (5.1)$$

$$\leq (1-\rho) \cdot |\eta_1 - \eta_2| + \quad (5.2)$$

$$\begin{aligned} &\mathbb{E}_{s' \sim P_{s,a}^\rho} [|(\eta_1 - V^*(s'))_+ - (\eta_2 - V^*(s'))_+|] \\ &\leq (2-\rho) \cdot |\eta_1 - \eta_2|. \end{aligned} \quad (5.3)$$

Here, Inequality (5.1) follows from Jensen's inequality, and (5.3) follows from the inequality $(a)_+ - (b)_+ \leq (a-b)_+$. Therefore, $F_{s,a}^\rho(\eta)$ is $(2-\rho)$ -Lipschitz continuous.

We now prove convexity. For any $t \in [0, 1]$, we have:

$$\begin{aligned} F_{s,a}^\rho(t\eta_1 + (1-t)\eta_2) &= \\ &\mathbb{E}_{s' \sim P_{s,a}^\rho} [(t\eta_1 + (1-t)\eta_2 - V^*(s'))_+] \end{aligned} \quad (5.4)$$

$$- (t\eta_1 + (1-t)\eta_2)(1-\rho) \quad (5.5)$$

$$\leq tF_{s,a}^\rho(\eta_1) + (1-t)F_{s,a}^\rho(\eta_2),$$

where Inequality (5.5) follows from the convexity of the function $(x)_+$ and the linearity of expectation.

Therefore, $F_{s,a}^\rho(\eta)$ is convex. The same argument applies to the empirical estimator $\hat{F}_{s,a}^\rho(\eta) = \frac{1}{C} \sum_{i=1}^C (\eta - V^*(s'_i))_+ - \eta(1-\rho)$, which is a finite sum of convex and $(2-\rho)$ -Lipschitz functions, and thus retains these properties. \blacksquare

Using Proposition 5.1.1, we derive concentration bounds that quantify how closely the SAA-based estimate $\hat{Q}^*(s, a)$, defined in (4.1), approximates the true robust action-value $Q^*(s, a)$ from (3.10). The proof leverages Hoeffding's inequality and the Lipschitz continuity properties from Proposition 5.1.1.

Lemma 5.1.2. *Let $\hat{Q}^*(s, a)$ denote the SAA estimate of the optimal robust action-value function as defined in Equation (4.1), and let $Q^*(s, a)$ denote the true optimal action-value function given in Equation (3.10). Then, for any $\lambda > 0$,*

$$\begin{aligned} \mathbb{P} \left(\left| \hat{Q}^*(s, a) - Q^*(s, a) \right| \geq \lambda \right) &\leq \\ &\frac{2}{\lambda} \left(\frac{8-4\rho}{(1-\gamma)\rho} \right) \cdot \exp \left(-\frac{C\lambda^2\rho^2(1-\gamma)^2}{2} \right). \end{aligned}$$

Proof. Let $\eta_{s,a}^*$ and $\hat{\eta}_{s,a}^*$ denote the optimal solutions to the stochastic optimization problems defined by $F_{s,a}^\rho(\eta)$ and $\hat{F}_{s,a}^\rho(\eta)$ in Equations (3.10) and (4.1), respectively. With this notation, we can express the true and SAA estimation of the robust action-

value functions as:

$$Q^*(s, a) = r(s, a) - \gamma F_{s,a}^\rho(\eta_{s,a}^*), \quad (5.6)$$

$$\hat{Q}^*(s, a) = r(s, a) - \gamma \hat{F}_{s,a}^\rho(\hat{\eta}_{s,a}^*). \quad (5.7)$$

Using the fact that $\hat{F}_{s,a}^\rho(\hat{\eta}_{s,a}^*) \leq \hat{F}_{s,a}^\rho(\eta_{s,a}^*)$, we have:

$$\begin{aligned} & \mathbb{P}(\hat{Q}^*(s, a) - Q^*(s, a) \geq \lambda) = \\ & \mathbb{P}(\hat{F}_{s,a}^\rho(\hat{\eta}_{s,a}^*) - F_{s,a}^\rho(\eta_{s,a}^*) \geq \lambda) \leq \\ & \mathbb{P}(\hat{F}_{s,a}^\rho(\eta_{s,a}^*) - F_{s,a}^\rho(\eta_{s,a}^*) \geq \lambda) = \\ & \mathbb{P}\left(\frac{1}{C} \sum_{i=1}^C (\eta_{s,a}^* - V^*(s'_i))_+ - \mathbb{E}_{s'}[(\eta_{s,a}^* - V^*(s'))_+] \geq \lambda\right). \end{aligned}$$

Applying Hoeffding's inequality (bounded by $\frac{2}{\rho(1-\gamma)}$):

$$\mathbb{P}(\hat{Q}^*(s, a) - Q^*(s, a) \geq \lambda) \leq \exp\left(-\frac{C\lambda^2(\rho(1-\gamma))^2}{2}\right).$$

For the lower tail, define a grid $\{\eta_i\}_{i=1}^d$ over $[0, \frac{2}{\rho(1-\gamma)}]$ with spacing $\frac{1}{d}$. Let:

$$A_N = \left\{ \max_{1 \leq i \leq d} (\hat{F}_{s,a}^\rho(\eta_i) - F_{s,a}^\rho(\eta_i)) < \frac{\lambda}{2} \right\}.$$

Then:

$$\mathbb{P}(A_N) \geq 1 - \mathbb{P}(\hat{F}_{s,a}^\rho(\eta_i) - F_{s,a}^\rho(\eta_i) \geq \frac{\lambda}{2}) \geq \quad (5.8)$$

$$1 - d \cdot \exp\left(-\frac{C\lambda^2(\rho(1-\gamma))^2}{8}\right), \quad (5.9)$$

where the last steps follows again by Hoeffding's inequality (bounded by $\frac{2}{\rho(1-\gamma)}$). Choose $d = \left\lceil \frac{1}{\lambda} \left(\frac{8-4\rho}{(1-\gamma)\rho} \right) \right\rceil$. Using Lipschitz continuity:

$$|\hat{F}_{s,a}^\rho(\eta_i) - \hat{F}_{s,a}^\rho(\hat{\eta}_{s,a}^*)| \leq (2-\rho) \cdot \frac{2}{\rho(1-\gamma)} \cdot \frac{1}{d} \leq \frac{\lambda}{2}. \quad (5.10)$$

This yields:

$$\mathbb{P}(Q^*(s, a) - \hat{Q}^*(s, a) < \lambda) = \quad (5.11)$$

$$\mathbb{P}(F_{s,a}^\rho(\eta_{s,a}^*) - \hat{F}_{s,a}^\rho(\hat{\eta}_{s,a}^*) < \lambda) \geq \quad (5.12)$$

$$1 - \frac{1}{\lambda} \left(\frac{8-4\rho}{(1-\gamma)\rho} \right) \cdot \exp\left(-\frac{C\lambda^2(\rho(1-\gamma))^2}{8}\right). \quad (5.13)$$

Combining both directions:

$$\mathbb{P} \left(\left| \hat{Q}^*(s, a) - Q^*(s, a) \right| \geq \lambda \right) \leq \quad (5.14)$$

$$\frac{2}{\lambda} \left(\frac{8 - 4\rho}{(1 - \gamma)\rho} \right) \cdot \exp \left(-\frac{C\lambda^2(\rho(1 - \gamma))^2}{2} \right). \quad (5.15)$$

It is important to note that the SAA-estimated robust action-value $\hat{Q}(s, a)$ is a theoretical construct, as the true robust value function $V^*(s')$ is not available in practice. In the RSS algorithm, at depth d , $V^*(s')$ is approximated by $\hat{V}_{d-1}(s')$, the estimated robust value of the sampled successor state from the previous depth. Although $\hat{Q}(s, a)$ is not computed directly during planning, it serves a key theoretical role in quantifying the error between the true robust action-value $Q^*(s, a)$ and the depth- d estimate $\hat{Q}_d(s, a)$ produced by RSS.

To capture how the estimation error accumulates across depths, we define a depth-dependent error term α_d recursively for given $\lambda > 0$:

$$\alpha_0 = 0, \quad \alpha_d = \gamma(\lambda + \alpha_{d-1}) \quad \text{for } H \geq d \geq 1. \quad (5.16)$$

This term quantifies the cumulative estimation error up to depth d . The error at depth d arises from two sources: the finite-sample approximation error λ and the accumulated error from the previous depth α_{d-1} . The combined error is $\lambda + \alpha_{d-1}$, discounted by γ , resulting in α_d . The following lemma shows that α_d serves as a probabilistic upper bound on the difference between $Q^*(s, a)$ and $\hat{Q}_d(s, a)$.

Lemma 5.1.3. *Let $Q^*(s, a)$ denote the optimal action-value function defined in Equation (3.10), and let $\hat{Q}_d(s, a)$ denote the estimate produced by the RSS algorithm at depth d , as defined in Equation (4.2). Then:*

$$\mathbb{P} \left(\left| \hat{Q}_d(s, a) - Q^*(s, a) \right| \leq \alpha_d \right) \geq 1 - (|A| \cdot C)^d \frac{2}{\lambda} \left(\frac{8 - 4\rho}{(1 - \gamma)\rho} \right) \cdot \exp \left(-\frac{C\lambda^2\rho^2(1 - \gamma)^2}{2} \right),$$

where $\alpha_0 = 0$ and $\alpha_d = \gamma(\lambda + \alpha_{d-1})$ for $H \geq d > 0$.

Proof. Let:

$$\begin{aligned} \eta_{s,a}^* &= \arg \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} F_{s,a}^\rho(\eta), \\ \hat{\eta}_{s,a}^* &= \arg \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} \hat{F}_{s,a}^\rho(\eta), \\ \tilde{\eta}_{s,a}^{*,d} &= \arg \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} \tilde{F}_{s,a}^{\rho,d}(\eta). \end{aligned}$$

We proceed by induction on the tree depth d .

For the base case $d = 0$, the bound holds trivially since $\alpha_0 = 0$ and Q^* is bounded by $\frac{1}{(1-\gamma)}$.

Assume the bound holds for depth $d - 1$. For depth d :

$$\begin{aligned} & |Q^*(s, a) - \hat{Q}_d(s, a)| \leq \\ & |Q^*(s, a) - \hat{Q}^*(s, a)| + |\hat{Q}^*(s, a) - \hat{Q}_d(s, a)|. \end{aligned}$$

We now bound each term separately.

For the first term:

$$Q^*(s, a) - \hat{Q}^*(s, a) = \tag{5.17}$$

$$F_{s,a}^\rho(\eta_{s,a}^*) - \hat{F}_{s,a}^\rho(\hat{\eta}_{s,a}^*) \leq \tag{5.18}$$

$$F_{s,a}^\rho(\hat{\eta}_{s,a}^*) - \hat{F}_{s,a}^\rho(\hat{\eta}_{s,a}^*) \leq \tag{5.19}$$

$$\gamma \left(\mathbb{E}_{s'}[(\hat{\eta}_{s,a}^* - V^*(s'))_+] - \frac{1}{C} \sum_{i=1}^C (\hat{\eta}_{s,a}^* - V^*(s'_i))_+ \right) \leq \gamma\lambda. \tag{5.20}$$

A symmetric argument yields:

$$\left| Q^*(s, a) - \hat{Q}^*(s, a) \right| \leq \gamma\lambda. \tag{5.21}$$

For the second term:

$$\hat{Q}^*(s, a) - \hat{Q}_d(s, a) = \tag{5.22}$$

$$\hat{F}_{s,a}^\rho(\hat{\eta}_{s,a}^*) - \tilde{F}_{s,a}^{\rho,d}(\tilde{\eta}_{s,a}^{*,d}) \leq \tag{5.23}$$

$$\hat{F}_{s,a}^\rho(\tilde{\eta}_{s,a}^{*,d}) - \tilde{F}_{s,a}^{\rho,d}(\tilde{\eta}_{s,a}^{*,d}) = \tag{5.24}$$

$$\gamma \left(\frac{1}{C} \sum_{i=1}^C (\tilde{\eta}_{s,a}^{*,d} - V(s'_i))_+ - (\tilde{\eta}_{s,a}^{*,d} - \hat{V}_{d-1}^*(s'_i))_+ \right) \leq \tag{5.25}$$

$$\gamma \left(\frac{1}{C} \sum_{i=1}^C (\hat{V}_{d-1}^*(s'_i) - V(s'_i))_+ \right) \leq \tag{5.26}$$

$$\gamma\alpha_d. \tag{5.27}$$

Therefore,

$$\left| Q^*(s, a) - \hat{Q}_d(s, a) \right| \leq \gamma\lambda + \gamma\alpha_d = \alpha_{d+1}. \tag{5.28}$$

The probability of a bad estimate compounds across K actions and C samples at each node, yielding a multiplicative factor $(KC)^d$. Applying the concentration bound from Lemma 5.1.2 completes the proof. \blacksquare

We now can recursively compute the bound α_H , which captures the total error at the root (depth H) between the optimal robust action-value function $Q^*(s, a)$ and the

RSS-estimated value $\hat{Q}_H(s, a)$.

$$\alpha_H = \left(\sum_{i=1}^H \gamma^i \lambda \right) + \gamma^H \frac{1}{1-\gamma} \leq \frac{1}{1-\gamma} (\lambda + \gamma^H). \quad (5.29)$$

By choosing the horizon $H = \lceil \log_\gamma(\lambda) \rceil$, we ensure with high probability that the RSS estimate at depth H satisfies $\alpha_H \leq \frac{2\lambda}{1-\gamma}$. Next, given any desired confidence level $1 - \delta$, where $0 < \delta < 1$ represents the maximum acceptable failure probability, we select a sufficiently large constant C to achieve this confidence. Specifically, we select C satisfying:

$$C \geq \frac{2}{\lambda^2 \rho^2 (1-\gamma)^2} \cdot \left(2H \ln \left(\frac{2|A| \cdot H}{\lambda^2 \rho^2 (1-\gamma)^2} \right) + \ln \left(\frac{2(8-4\rho)}{\delta \lambda (1-\gamma) \rho} \right) \right). \quad (5.30)$$

Using the parameters set above, we guarantee that, with probability at least $1 - \delta$, the RSS estimate at depth H is within $\frac{2\lambda}{1-\gamma}$ of the true robust action-value function. To complete our analysis, we present the following lemma, which relates the robust action-value function to the robust value function. This result holds for any stochastic policy and extends the original lemma from [9].

Lemma 5.1.4. *Denote π^* as the optimal robust policy. Let π be a stochastic policy such that $\mathbb{P}(Q^*(s, \pi^*(s)) - Q^*(s, \pi(s)) < \lambda) \geq 1 - \delta$ for all s . Then:*

$$V^*(s) - V^\pi(s) \leq \frac{\delta}{(1-\gamma)^2} + \frac{\lambda}{1-\gamma}. \quad (5.31)$$

Proof. Since the reward is bounded by $0 \leq r \leq 1$, we have that:

$$\begin{aligned} \mathbb{E}_{a \sim \pi(s)}[Q^*(s, a)] &\geq \\ (1-\delta)(Q^*(s, \pi^*(s)) - \lambda) &\geq \\ Q^*(s, \pi^*(s)) - \delta \frac{1}{1-\gamma} - \lambda. & \end{aligned}$$

Now denote $\beta = \frac{\delta}{1-\gamma} + \lambda$, thus we have that $Q^*(s, \pi^*(s)) - Q^*(s, \pi(s)) \leq \beta$ which implies $|\mathbb{E}[r(s, \pi^*(s))] - \mathbb{E}[r(s, \pi(s))]| \leq \beta$. Now consider a policy π^j that executes π

for the first $j + 1$ steps and then executes $\pi^*(s)$ for the rest of the time. Thus, we have

$$\begin{aligned}
V^*(s) - V^{\pi^j}(s) &= \\
\min_{P' \in \mathcal{P}} V^{\pi^*, P'}(s) - \min_{P' \in \mathcal{P}} V^{\pi^j, P'}(s) &= \\
V^{\pi^*, P^*}(s) - V^{\pi^j, P^*}(s) &\leq \\
V^{\pi^*, P^*}(s) - V^{\pi^j, P^*}(s) &\leq \\
\sum_{i=0}^j \beta \gamma^i. &
\end{aligned}$$

Thus, we have that: $V^*(s) - V^\pi(s) \leq \frac{\beta}{1-\gamma} = \frac{\delta}{(1-\gamma)^2} + \frac{\lambda}{1-\gamma}$. ■

Leveraging Lemma 5.1.3, we see that the RSS stochastic policy π meets the condition required by Lemma 5.1.4. Applying this lemma directly to the RSS policy, we obtain a bound relating the RSS policy's value function to the true robust optimal value function $V^*(s)$. This argument culminates in our main theoretical result in theorem 5.1.

Theorem 5.2 (Theorem 5.1). *For any $s \in \mathcal{S}$ and any $\epsilon > 0$, the Robust Sparse Sampling algorithm returns a policy π such that:*

$$|V^\pi(s) - V^*(s)| \leq \epsilon,$$

with the following hyperparameters:

$$\begin{aligned}
\lambda &= \frac{\epsilon}{3}, \quad \delta = \lambda(1 - \gamma), \\
H &= \lceil \log_\gamma(\lambda) \rceil, \\
C &= \frac{2}{\lambda^2 \rho^2 (1 - \gamma)^2} \cdot \\
&\left(2H \ln \left(\frac{2KH}{\lambda^2 \rho^2 (1 - \gamma)^2} \right) + \ln \left(\frac{2(8 - 4\rho)}{\delta \lambda (1 - \gamma) \rho} \right) \right).
\end{aligned}$$

Proof. By Lemma 5.1.3 we have that the error in the estimation of Q^* is at most α_H , with probability $1 - (KC)^d \frac{2}{\lambda} \left(\frac{8-4\rho}{(1-\gamma)\rho} \right) \cdot \exp \left(-\frac{C\lambda^2(\rho(1-\gamma))^2}{2} \right)$. Using the values that appears in the theorem for C and H we have that with probability $1 - \delta$ the error is at most $\frac{2\lambda}{1-\gamma}$. Thus, we can apply Lemma 5.1.4 to conclude that the policy π computed by the Robust Sparse Sampling algorithm is such that for every state s we have that:

$$V^*(s) - V^\pi(s) \leq \frac{2\lambda}{(1-\gamma)} + \frac{\delta}{(1-\gamma)^2}.$$

Substituting $\delta = \lambda(1 - \gamma)$ and $\lambda = \frac{\epsilon}{3}$ we have that:

$$V^*(s) - V^\pi(s) \leq \epsilon.$$

5.2 Computational Complexity

The RSS algorithm builds a lookahead tree of depth H , where each node branches into $|A| \cdot C$ children—corresponding to $|A|$ actions and C sampled next states per action. Therefore, the total number of nodes in the tree is $(|A| \cdot C)^H$.

At each node, the algorithm performs two main operations: (1) sampling C next states using the generative model, and (2) solving the SAA optimization problem defined in Equation 4.2. The sampling step incurs a cost of $O(C)$. For the optimization step, the algorithm minimizes a piecewise-linear convex function $\tilde{F}_{s,a}^{\rho,d}(\eta)$, which has C breakpoints at the values $\{\hat{V}_{d-1}(s'_i)\}_{i=1}^C$. The minimum is guaranteed to lie at one of the breakpoints or at the boundary points $\eta = 0$ and $\eta = \frac{2}{\rho(1-\gamma)}$. The optimal solution can thus be found by first sorting the C breakpoints in $O(C \log C)$ time, followed by a linear scan to identify the minimizer, resulting in a total per-node complexity of:

$$O(C \log C).$$

Multiplying this by the total number of nodes yields the overall computational complexity of RSS:

$$O\left((|A| \cdot C \log C)^H\right).$$

For comparison, the standard Sparse Sampling algorithm has complexity $O\left((|A| \cdot C)^H\right)$, implying that RSS introduces only an additional logarithmic factor due to the robust optimization step. Importantly, in most practical applications, sampling successor states from the generative model dominates the computational cost. As a result, the added $\log C$ factor in RSS is typically negligible in practice and does not significantly affect overall runtime.

Chapter 6

Experiments

We evaluate the performance of the proposed RSS algorithm in two benchmark environments: **FrozenLake** and **CartPole**, aiming to empirically assess its robustness under model misspecification and compare it to standard Sparse Sampling (SS).

All experiments are conducted in the setting of *online planning with model uncertainty*. The agent computes actions by simulating future trajectories using an inaccurate generative model, differing from the true environment dynamics.

In these environments, uncertainty is present only in certain regions, while others are accurately modeled. This reflects a common real-world scenario in which hazardous or rarely visited states lack sufficient data, resulting in higher model uncertainty, whereas frequently visited safe regions benefit from more reliable transition estimates. To capture this structure, we apply the robust backup update (4.2) exclusively in states with uncertainty. In all other states, we use the standard expected backup (3.11). Full algorithmic details are provided in the supplementary material.

This selective use of robust backups preserves the theoretical guarantees established in Theorem 5.1, while avoiding overly conservative behavior in well-modeled regions. Further discussion of this design choice appears in the supplementary material.

6.1 FrozenLake

Environment. The **FrozenLake** task is played on an 8×8 grid. The agent begins in the upper-left cell and must navigate to the goal in the lower-right cell without falling into any of the "hole" cells scattered throughout the grid. At each time step, the agent chooses one of four actions: **up**, **down**, **left** or **right**, but movement is stochastic: with probability p the agent moves in the intended direction, and with probability $(1-p)/2$ it instead slips to one of the two orthogonal neighbors.

The immediate reward at each state is defined as: $r(s) = \frac{1}{(d(s)+1)^3}$, where $d(s)$ is the Manhattan distance from state s to the goal. A terminal reward of 1 is granted upon reaching the goal, while falling into a hole yields a reward of 0. Each episode ends when a terminal state is reached or after 150 time steps.

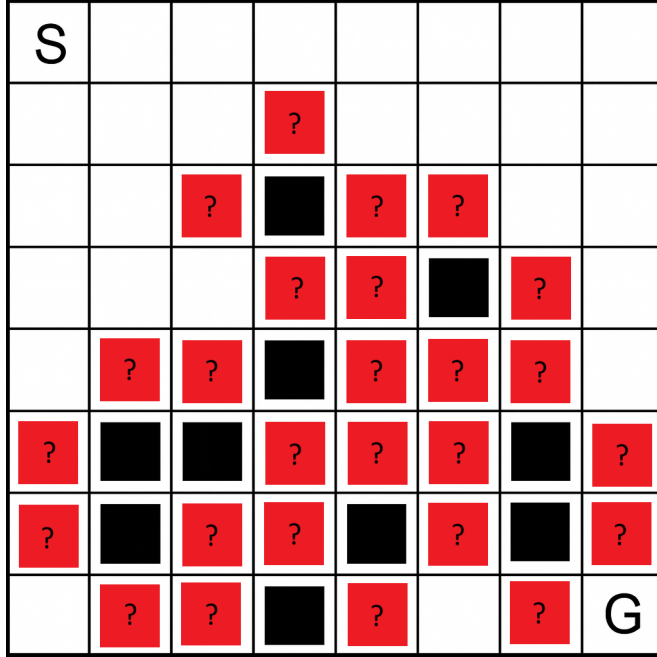


Figure 6.1: Visualization of the 8×8 FrozenLake environment, a stochastic grid-world where the agent starts in the top-left cell (S) and aims to reach the goal in the bottom-right cell (G), while avoiding hazardous holes represented by black squares. Due to the stochastic nature of the environment, the agent’s actions may not always result in the intended direction. Cells adjacent to holes are marked with red squares containing question marks, highlighting regions of model uncertainty where the agent’s planning model deviates from the true environment dynamics.

Model Uncertainty. In our setup, the true transition dynamics are defined using $p = 0.4$. However, the agent plans using an approximate model that differs only in states adjacent to holes. In these uncertain regions, the probability of moving in the intended direction is increased to $p^\rho = p + \rho$, while the probabilities of deviating to either perpendicular direction are adjusted to $(1 - p^\rho)/2$. This modification satisfies the uncertainty condition defined in (3.1). Elsewhere, the approximate model matches the true dynamics exactly. A visualization of the environment is shown in Figure 6.1.

Experimental Setup and Results. Experimental results are summarized in Table 6.1. We evaluate RSS and standard SS under varying levels of model uncertainty, each over 1000 different seeds. As a benchmark, we also evaluate SS with full access to the true dynamics, achieving an average discounted return of 0.249 ± 0.012 . All methods use a planning horizon of $H = 3$, sample width $C = 50$, and discount factor $\gamma = 0.99$. The uncertainty budget ρ is varied across the set $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$.

As expected, both RSS and SS underperform compared to the SS variant with full access to the true environment dynamics. However, RSS consistently demonstrates better performance than SS across all values of ρ , with the performance gap widening as uncertainty increases. This highlights RSS’s robustness to model misspecification

and its ability to maintain stronger performance under growing uncertainty.

| ρ | RSS | SS |
|--------|--------------------------|-------------------|
| 0.1 | 0.177 \pm 0.011 | 0.172 \pm 0.011 |
| 0.2 | 0.171 \pm 0.011 | 0.123 \pm 0.009 |
| 0.3 | 0.145 \pm 0.010 | 0.109 \pm 0.009 |
| 0.4 | 0.126 \pm 0.009 | 0.098 \pm 0.008 |
| 0.5 | 0.127 \pm 0.009 | 0.080 \pm 0.007 |
| 0.6 | 0.118 \pm 0.009 | 0.080 \pm 0.008 |

Table 6.1: Performance of RSS and SS in the FrozenLake environment under varying uncertainty levels ρ . The reported values are the average discounted return with the standard error over 1000 different seed. The best-performing algorithm for each ρ is highlighted in bold. The average discounted return of SS with access to the true dynamics is 0.249 ± 0.012 .

6.2 CartPole

Environment. We use the `CartPole` environment, where the agent must balance a pole on a moving cart by applying discrete left or right forces. The continuous state is defined by the cart’s position x , velocity \dot{x} , pole angle θ , and angular velocity $\dot{\theta}$. An episode terminates if $|\theta| > 0.2$ radians, $|x| > 2.4$, or after 200 time steps. The reward is defined as $r(\theta) = 1 - 0.2|\theta|$ in non-terminal states, and 0 otherwise, encouraging the pole to remain upright.

At the start of each episode, the cart is centered and the pole is vertical. At each step, the agent selects a force, transitioning to the next state according to deterministic dynamics, with added Gaussian noise $\mathcal{N}(0, \sigma_\theta^2(x))$ on the pole angle. The noise variance depends on the current cart position x , defined as:

$$\sigma_\theta^2(x) = \begin{cases} \sigma_{high}^2, & \text{if } x_a < |x| < x_b \\ \sigma_{low}^2, & \text{otherwise} \end{cases} \quad (6.1)$$

This models a narrow “hazard zone” $x \in \pm[x_a, x_b]$, where the system is more unstable due to higher noise.

Model Uncertainty. The hazard zone is assumed to be narrow and difficult to model accurately. As such, the planning model assumes a constant low noise σ_{low}^2 across all states, underestimating the true noise in the hazard zone. This mismatch induces localized model uncertainty. Full noise specifications and uncertain total variance calculations are detailed in the supplementary material.

Experimental Setup and Results. We set $x_a = 0.02$, $x_b = 0.03$, $\sigma_{low} = 10^{-3}$, and vary σ_{high} from 0.07 to 0.15. We compare RSS against standard SS, using a planning

horizon $H = 5$, width $C = 10$, and discount factor $\gamma = 0.999$. As a reference, we also evaluate SS with access to the true noise model. Each configuration is averaged over 500 random seeds.

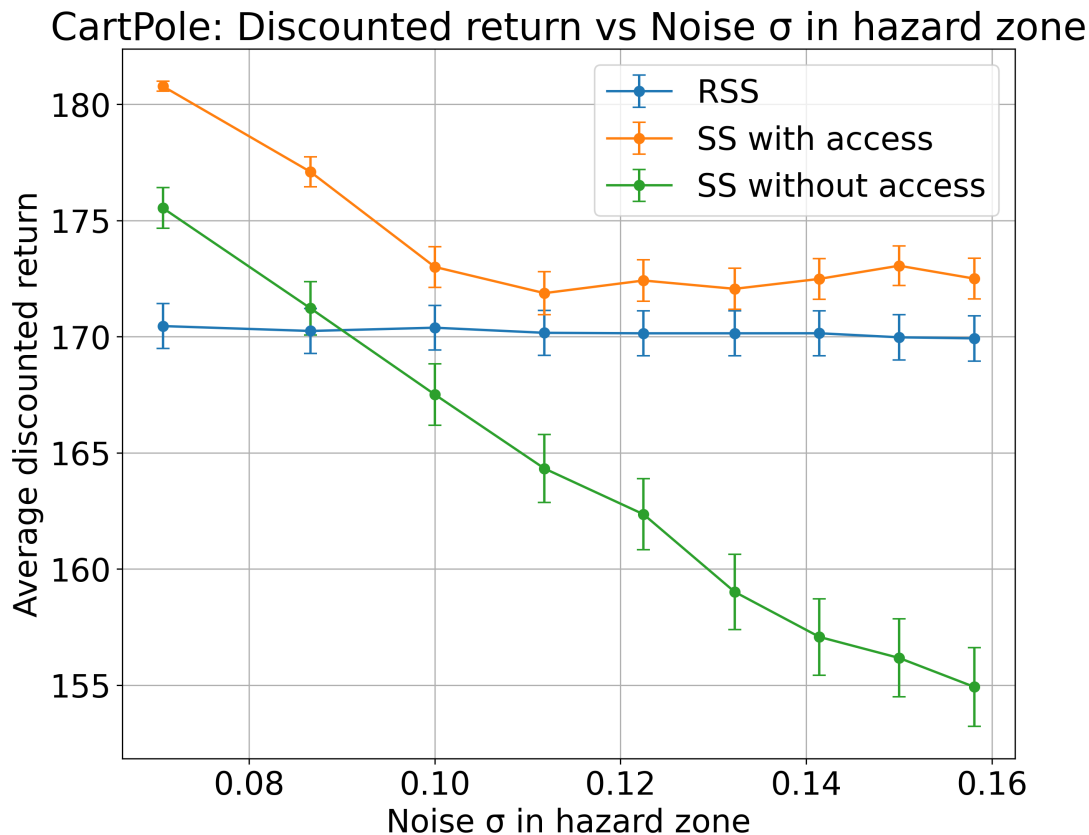


Figure 6.2: Average discounted return comparison of RSS and SS (with/without access to true dynamics) under increasing noise variance in the hazard zone. Error bars denote standard error across 500 different seeds.

Figures 6.2 show the average discounted performance as a function of the noise standard deviation σ_{high} within the hazard zone. An additional figure in the supplementary material presents the success rate—defined as completing 200 steps without termination—under varying noise levels. As expected, the SS variant with full access to the true dynamics achieves the highest performance across all noise levels, as it can plan optimally using accurate environment information. The performance of both SS variants (with and without model access) degrades as noise increases, indicating their increased sensitivity to unmodeled uncertainty.

In contrast, RSS maintains stable performance across all levels of noise, demonstrating its robustness to model misspecification. Notably, in low-noise regimes, RSS underperforms compared to SS without access. This is a known phenomenon in robust planning: robust policies are inherently conservative, as they optimize for the worst-case plausible dynamics within an uncertainty set, leading to overly cautious behavior that sacrifices performance for safety [17].

However, as the noise variance increases, RSS maintains a near-constant performance level in this scenario. SS without access to the true model continues to rely on an underestimated noise model, resulting in unsafe and suboptimal actions, while RSS anticipates and mitigates adverse dynamics. As a result, RSS eventually outperforms SS without access in both return and success rate. This crossover point highlights the fundamental trade-off in robust planning: while robust methods may underperform in low-risk settings, they provide significant benefits in high-uncertainty environments by reducing risk and failure rates.

Chapter 7

Conclusions

In this work, we introduced the Robust Sparse Sampling (RSS) algorithm, the first online planning algorithm for RMDPs with finite-sample theoretical performance guarantees. RSS extends the Sparse Sampling algorithm by incorporating robustness against model errors, leveraging the dual formulation of robust value functions and Sample Average Approximation (SAA) techniques. Our theoretical analysis establishes finite-time error bounds for RSS, and we demonstrate its effectiveness in simulative experiments in environments with uncertain transition dynamics.

We hope that the RSS algorithm and methods will serve as a foundation for future research in robust online planning, both for methods that can scale better for large state and action spaces, and for online methods that can handle model uncertainty. Moreover, we wish to extend our methods to anytime-fashion Monte Carlo Tree Search (MCTS), and to Partially Observable Markov Decision Process (POMDP) settings.

Limitations While the RSS algorithm is the first to address robust online planning under model uncertainty with formal performance guarantees, it exhibits several important limitations.

Similar to Sparse Sampling, RSS suffers from significant sample and computational inefficiency, as the complexity grows exponentially with the planning horizon H . This severely restricts its practical applicability in environments requiring long-horizon planning. However, in settings where short planning horizons are sufficient—e.g., due to low discount factors or inherently short episodes—RSS may still offer a viable and effective alternative.

Second, the algorithm assumes prior knowledge of the uncertainty budget parameter ρ . In real-world applications, accurately estimating ρ is often nontrivial, especially in non-stationary or partially observed environments where the transition dynamics may evolve over time. Estimating such parameters reliably remains an open problem in robust decision-making [13, 31].

A current limitation that leads to over-conservatism is the rectangularity assumption of the uncertainty set. Recent works have shown promising directions to address

this issue [6], and we hope to incorporate those in online robust planning as well.

Appendix A

Appendix

A.1 Further Implementation Details

A.1.1 RSS Implementation with State-Action Dependent ρ

In this section, we describe the implementation details of the RSS algorithm when the robustness parameter ρ is not constant, but varies with the state and action, i.e., $\rho = \rho(s, a) \in [0, 1]$. This setting is more realistic in many practical applications, where certain regions of the environment are well understood due to ample data, while other regions remain uncertain because they are rarely visited.

Using the robust backup update indiscriminately across all states and actions can result in overly conservative behavior, particularly in areas where the model is already accurate. To avoid this, we introduce a modification to the RSS algorithm, provided in Algorithm A.1. The algorithm behaves similarly to the original RSS, but applies the robust backup update only in state-action pairs where $\rho(s, a) > 0$. In all other cases, it defaults to the standard expected backup. This ensures the algorithm remains cautious in uncertain regions while avoiding unnecessary conservatism in well-understood parts of the environment.

From a theoretical standpoint, the guarantees in Theorem 5.1 still hold with minor adjustments. Let $\rho_{\max} = \max_{s,a} \rho(s, a)$. By replacing the constant ρ in the original proofs with ρ_{\max} , we can extend the results of Lemmas 5.1.2, 5.1.3, and 5.1.4 to this variable- ρ setting. Consequently, the bound on the value function remains valid, though it will be scaled by ρ_{\max} instead of the specific values $\rho(s, a)$. While this substitution may introduce looseness in regions where $\rho(s, a) \ll \rho_{\max}$, the overall theoretical structure remains intact.

A.1.2 CartPole Environment

In this section, we provide a detailed description of the CartPole environment used in our experiments, focusing on how model uncertainty is calculated.

The CartPole standard dynamics are governed by deterministic second-order dif-

Algorithm A.1 Robust Sparse Sampling (RSS) - Changing ρ

Input: Current state s , current depth d

Parameter: Sample width C , planning horizon H computed based on Theorem 5.1

Output: Estimated optimal action and its value

```
1: if  $d = 0$  then
2:   return 0
3: end if
4: for all  $a \in \mathcal{A}$  do
5:    $V_{\text{list}} \leftarrow []$ 
6:   for  $i = 1$  to  $C$  do
7:     Sample  $s'_i \sim P^o(\cdot | s, a)$ 
8:      $(\_, \hat{V}_{d-1}(s'_i)) \leftarrow \text{RSS}(s'_i, d - 1)$ 
9:     Append  $\hat{V}_{d-1}(s'_i)$  to  $V_{\text{list}}$ 
10:  end for
11: end for
12: if  $\rho(s, a) > 0$  then
13:   Update  $\hat{Q}_d(s, a)$  using Equation (4.2) with  $V_{\text{list}}$  and  $\rho(s, a)$ 
14: else
15:   Update  $\hat{Q}_d(s, a)$  using Equation (3.11) with  $V_{\text{list}}$ 
16: end if
17: return  $\arg \max_{a \in \mathcal{A}} \hat{Q}_d(s, a), \max_{a \in \mathcal{A}} \hat{Q}_d(s, a)$ 
```

ferential equations. Let the system's state be defined by the cart's position x , velocity \dot{x} , the pole's angle θ , and angular velocity $\dot{\theta}$, with a control action a . The next state is determined by a transition function $f(x, \dot{x}, \theta, \dot{\theta}, a)$, such that:

$$\begin{pmatrix} x' \\ \dot{x}' \\ \theta' \\ \dot{\theta}' \end{pmatrix} = f(x, \dot{x}, \theta, \dot{\theta}, a). \quad (\text{A.1})$$

In our stochastic setting, we introduce model uncertainty by injecting Gaussian noise into the pole angle after applying the action. Specifically, the next state becomes:

$$\begin{pmatrix} x' \\ \dot{x}' \\ \theta' + \epsilon \\ \dot{\theta}' \end{pmatrix} = f(x, \dot{x}, \theta, \dot{\theta}, a) + \begin{pmatrix} 0 \\ 0 \\ \epsilon \\ 0 \end{pmatrix}, \quad \epsilon \sim \mathcal{N}(0, \sigma_\theta^2(x)) \quad (\text{A.2})$$

This results in a stochastic transition model where noise is added only to the pole angle. The transition distribution is given by:

$$P(\tilde{x}, \tilde{\dot{x}}, \tilde{\theta}, \tilde{\dot{\theta}} | x, \dot{x}, \theta, \dot{\theta}, a) = \delta((\tilde{x}, \tilde{\dot{x}}, \tilde{\theta}) - (x', \dot{x}', \theta')) \mathcal{N}(\tilde{\theta}; \theta', \sigma_\theta^2(x)) \quad (\text{A.3})$$

In regions of the state space considered safe, both the true and estimated models use the same noise level σ_{low}^2 , and hence their total variation (TV) distance is zero. In contrast, in uncertain or “dangerous” regions, the true transition model uses a higher variance σ_{high}^2 , while the agent underestimates the noise using σ_{low}^2 . The total variation distance between these two distributions is:

$$\begin{aligned} & \text{TV}(\mathcal{N}(\mu, \sigma_{low}^2), \mathcal{N}(\mu, \sigma_{high}^2)) = \\ & \text{erf}\left(\frac{\sigma_{high}\sqrt{\ln(\sigma_{high}/\sigma_{low})}}{\sqrt{2(\sigma_{high}^2 - \sigma_{low}^2)}}\right) - \text{erf}\left(\frac{\sigma_{low}\sqrt{\ln(\sigma_{high}/\sigma_{low})}}{\sqrt{2(\sigma_{high}^2 - \sigma_{low}^2)}}\right) \end{aligned} \quad (\text{A.4})$$

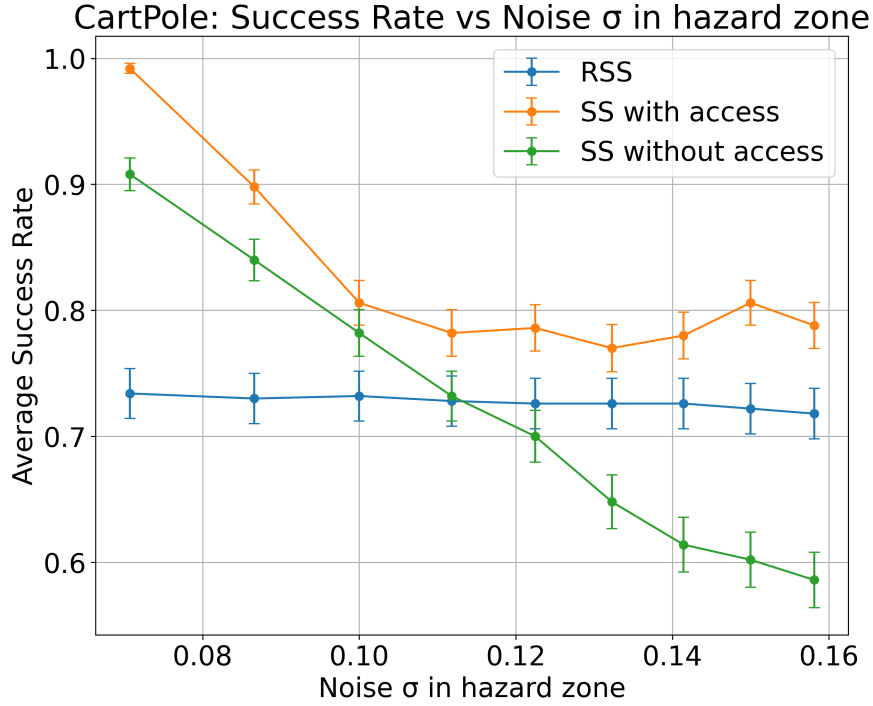


Figure A.1: Average success rate (i.e., completing 200 steps without termination) comparison of RSS and SS (with/without access to true dynamics) under increasing noise variance in the hazard zone. Error bars denote standard error across 500 different seeds.

Bibliography

- [1] Daniel Berend and Aryeh Kontorovich. On the convergence of the empirical distribution. *arXiv preprint arXiv:1205.6711*, 2012.
- [2] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.
- [3] Javier Burroni, Justin Domke, and Daniel Sheldon. Sample average approximation for black-box vi. *arXiv preprint arXiv:2304.06803*, 2023.
- [4] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *International conference on computers and games*, pages 72–83. Springer, 2006.
- [5] Jing Dong, Jingwei Li, Baoxiang Wang, and Jingzhao Zhang. Online policy optimization for robust mdp. *arXiv preprint arXiv:2209.13841*, 2022.
- [6] Vineet Goyal and Julien Grand-Clement. Robust markov decision processes: Beyond rectangularity. *Mathematics of Operations Research*, 48(1):203–226, 2023.
- [7] Klein Haneveld and Maarten H Van der Vlerk. *Stochastic programming*. Springer, 2020.
- [8] Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.
- [9] Michael Kearns, Yishay Mansour, and Andrew Y. Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine Learning*, 49(2-3):193–208, 2002.
- [10] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.
- [11] Sven Koenig. Agent-centered search. *AI Magazine*, 22(4):109–109, 2001.

- [12] Farnaz Kohankhaki, Kiarash Aghakasiri, Hongming Zhang, Ting-Han Wei, Chao Gao, and Martin Müller. Monte carlo tree search in the presence of transition uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20151–20158, 2024.
- [13] Navdeep Kumar, Kaixin Wang, Uri Gadot, Kfir Yehuda Levy, and Shie Mannor. Learning the uncertainty set in robust markov decision process. In *The Second Tiny Papers Track at ICLR 2024*, 2024.
- [14] Michael H Lim, Tyler J Becker, Mykel J Kochenderfer, Claire J Tomlin, and Zachary N Sunberg. Optimality guarantees for particle belief approximation of pomdps. *Journal of Artificial Intelligence Research*, 77:1591–1636, 2023.
- [15] Michael H Lim, Claire J Tomlin, and Zachary N Sunberg. Sparse tree search optimality guarantees in pomdps with continuous observation spaces. *arXiv preprint arXiv:1910.04332*, 2019.
- [16] Michael L. Littman, Thomas L. Dean, and Leslie Pack Kaelbling. On the complexity of solving markov decision problems. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI’95, pages 394–402, 1995.
- [17] Shie Mannor, Ofir Mebel, and Huan Xu. Lightning does not strike twice: Robust mdps with coupled uncertainty. *arXiv preprint arXiv:1206.4643*, 2012.
- [18] Shie Mannor, Duncan Simester, Peng Sun, and John N Tsitsiklis. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.
- [19] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.
- [20] Kishan Panaganti and Dileep Kalathil. Sample complexity of robust reinforcement learning with a generative model. In *International Conference on Artificial Intelligence and Statistics*, pages 9582–9602. PMLR, 2022.
- [21] Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. Robust reinforcement learning using offline data, 2022.
- [22] Christos H Papadimitriou and John N Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.
- [23] Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.
- [24] Stéphane Ross, Joelle Pineau, Sébastien Paquet, and Brahim Chaib-Draa. On-line planning algorithms for pomdps. *Journal of Artificial Intelligence Research*, 32:663–704, 2008.

- [25] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2021.
- [26] Alexander Shapiro and Yan Li. Risk-averse formulations of stochastic optimal control and markov decision processes. *arXiv preprint arXiv:2505.16651*, 2025.
- [27] Apoorva Sharma, James Harrison, Matthew Tsao, and Marco Pavone. Robust and adaptive planning under model uncertainty. In *Proceedings of the international conference on automated planning and scheduling*, volume 29, pages 410–418, 2019.
- [28] David Silver and Joel Veness. Monte-carlo planning in large pomdps. *Advances in neural information processing systems*, 23, 2010.
- [29] Devang Sinha and Siddhartha P Chakrabarty. Multilevel monte carlo in sample average approximation: Convergence, complexity and application. *arXiv preprint arXiv:2407.18504*, 2024.
- [30] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. Despot: Online pomdp planning with regularization. *Advances in neural information processing systems*, 26, 2013.
- [31] Marnix Suilen, Thiago D Simão, David Parker, and Nils Jansen. Robust anytime learning of markov decision processes. *Advances in Neural Information Processing Systems*, 35:28790–28802, 2022.
- [32] Aviv Tamar, Shie Mannor, and Huan Xu. Scaling up robust mdps using function approximation. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 181–189, Beijing, China, June 2014. PMLR.
- [33] Yue Wang and Shaofeng Zou. Online robust reinforcement learning with model uncertainty. *Advances in Neural Information Processing Systems*, 34:7193–7206, 2021.

(Model uncertainty)(MCTS)Sparse Sampling(Sample-based methods)(MDPs)(Online planning)

(Finite-sample guarantees)RMDPs*Robust Sparse Sampling (RSS)*(Model uncertainty)(RMDPs)

(Sample Average Approxima-(RMDPs)Robust Sparse Sampling (RSS)Sparse Sampling
Sparse SamplingRSSRSS(tractable)tion - SAA)

Tamir Shazman, Idan Lev-Yehudi, Ron Benchetrit, and Vadim Indelman. Online robust planning under model uncertainty: A sample-based approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2026.

